

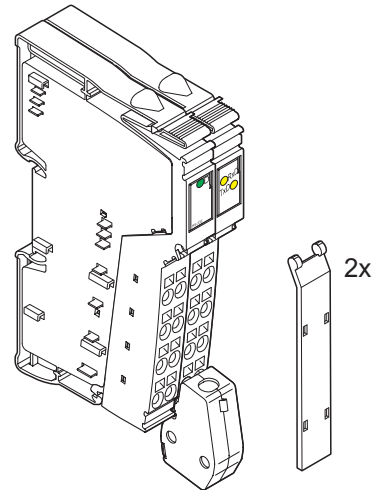
IB IL RS 232-PRO (-PAC)

Inline terminal for serial data transmission;
firmware version 1.20 or later

AUTOMATION

Data sheet
7112_en_03

© PHOENIX CONTACT - 10/2008



1 Function description

The terminal is designed for use within an Inline station. It is used to operate standard I/O devices with serial interfaces on a bus system.

Difference between IB IL RS 232 and IB IL RS 232-PRO

IB IL RS 232

Parameterization and data exchange are carried out via the bus using PCP services.

IB IL RS 232-PRO

Parameterization and data exchange are carried out via the bus using process data. This makes faster communication times possible with smaller amounts of data than with the IB IL RS 232.

Features

- A serial I/O channel (V.24 (RS-232))
- DTR/CTS handshake supported
- Various protocols supported
- Transmission speed can be set up to 38400 baud
- Number of data bits, stop bits, and parity can be set
- 4 kbyte receive buffer and 1 kbyte transmit buffer
- Parameterization and data exchange via the bus using process data
- Diagnostic and status indicators



This data sheet is only valid in association with the IL SYS INST UM E user manual.



Make sure you always use the latest documentation.
It can be downloaded at www.download.phoenixcontact.com.



This data sheet is valid for the products listed on the following page:

2 Ordering data

Products

Description	Type	Order No.	Pcs./Pkt.
Inline terminal for serial data transmission, including connectors and labeling fields	IB IL RS 232-PRO-PAC	2878722	1
Inline terminal for serial data transmission	IB IL RS 232-PRO	2878515	1



The listed connector set is needed for the complete fitting of the IB IL RS 232-PRO terminal.

Accessories

Description	Type	Order No.	Pcs./Pkt.
Connector set with a standard connector and a shield connector	IB IL AO/CNT-PLSET	2732664	1

Documentation

Description	Type	Order No.	Pcs./Pkt.
"Automation terminals of the Inline product range" user manual	IL SYS INST UM E	2698737	1
"INTERBUS addressing" data sheet	DB GB IBS SYS ADDRESS	9000990	1

3 Technical data

General data

Housing dimensions (width x height x depth)	24.4 mm x 136 mm x 72 mm (with connectors)
Weight	90 g (without connectors), 135 g (with connectors)
Operating mode	Process data mode with 6 words
Transmission speed	500 kbps
Permissible temperature (operation)	-25°C to +55 °C
Permissible temperature (storage/transport)	-25°C to +85 °C
Permissible humidity (operation/storage/transport)	10% to 95% according to DIN EN 61131-2
Permissible air pressure (operation/storage/transport)	70 kPa to 106 kPa (up to 3000 m above sea level)
Degree of protection	IP20
Class of protection	Class 3 according to EN 61131-2, IEC 61131-2
Connection data for Inline connectors	
Connection method	Spring-cage terminals
Conductor cross-section	0.08 mm ² to 1.5 mm ² (solid or stranded), 28 - 16 AWG

Interfaces

Bus

Local bus	Through data routing
-----------	----------------------

Serial interfaces

Type	V.24 (RS-232) interface with DTR/CTS handshake Data terminal equipment (DTE) Electrical data according to EIA (RS) 232, CCITT V.28, DIN 66259 Part 1
Input impedance	5 kΩ typical
Permissible input voltage range	-30 V to +30 V
Switching thresholds	0.8 V to 2.4 V
Hysteresis	0.5 V typical
Output voltage "HIGH" (with 3 kΩ load)	6.7 V typical
Output voltage "LOW" (with 3 kΩ load)	-6.7 V typical
Output voltage "HIGH" (no-load operation)	≤25 V
Output voltage "LOW" (no-load operation)	≥ -25 V

Interfaces (continued)

Permissible load capacity	2500 pF
Short-circuit protected against GND	Yes
Short-circuit current	±60 mA, maximum

Power consumption

Communications power U_L	7.5 V
Current consumption at U_L	155 mA, typical, 225 mA, maximum*
Total power consumption	1.163 W, typical; 1.688 W, maximum (approximately)*

* All connections of the serial interface are short-circuited.



This terminal takes no current from the U_M and U_S potential jumpers.

Supply of the module electronics by the bus terminal module

Connection method	Potential routing
-------------------	-------------------

Power dissipation

Power dissipation in the module	$P_{EL} = 1.163 \text{ W}$
Power dissipation of the housing P_{HOU}	1.2 W, maximum (within the permissible operating temperature)

Limitation of simultaneity, derating

No limitation of simultaneity, no derating

Protective equipment

None

Electrical isolation/isolation of the voltage areas

Electrical isolation of the logic level from the serial interface is ensured by the DC/DC converter.

Common potentials

The serial interface control and data lines have galvanically the same potential. FE is a separate potential area.

Separate potentials in the system consisting of bus terminal/power terminal and I/O terminal

Test distance	Test voltage
5 V supply incoming remote bus/7.5 V supply (bus logic)	500 V AC, 50 Hz, 1 min
5 V supply outgoing remote bus/7.5 V supply (bus logic)	500 V AC, 50 Hz, 1 min
V.24 (RS-232) interface/7.5 V supply (bus logic)	500 V AC, 50 Hz, 1 min
V.24 (RS-232) interface/24 V supply (I/O)	500 V AC, 50 Hz, 1 min
V.24 (RS-232) interface/functional earth ground	500 V AC, 50 Hz, 1 min
7.5 V supply (bus logic) / 24 V supply (I/O)	500 V AC, 50 Hz, 1 min
7.5 V supply (bus logic) / functional earth ground	500 V AC, 50 Hz, 1 min
24 V supply (I/O) / functional earth ground	500 V AC, 50 Hz, 1 min

Error messages to the higher-level control or computer system

None

Approvals

For the latest approvals, please visit www.download.phoenixcontact.com or www.eshop.phoenixcontact.com.

4 Diagnostic/status indicators and terminal point assignment

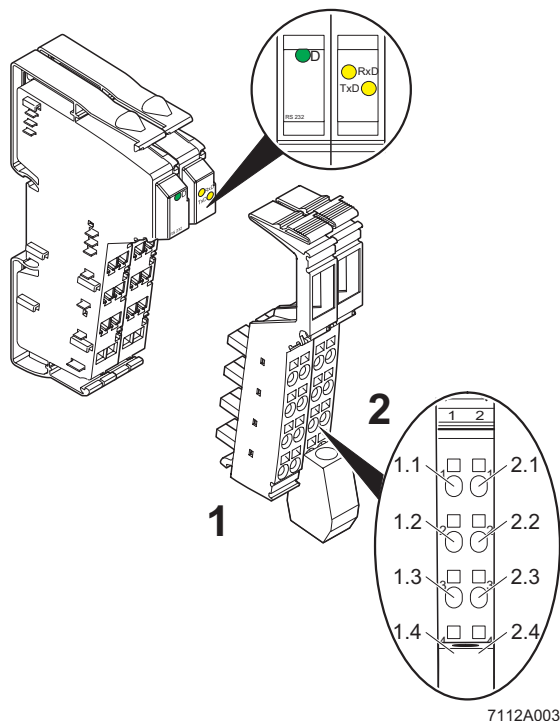


Figure 1 IB IL RS 232-PRO-PAC

4.1 Local diagnostic and status indicators

Des.	Color	Meaning
D	Green	Diagnostics
Serial interface:		
RxD	Yellow	Terminal is receiving data from the connected device
TxD	Yellow	Terminal is transmitting data to the connected device

4.2 Function identification

Orange

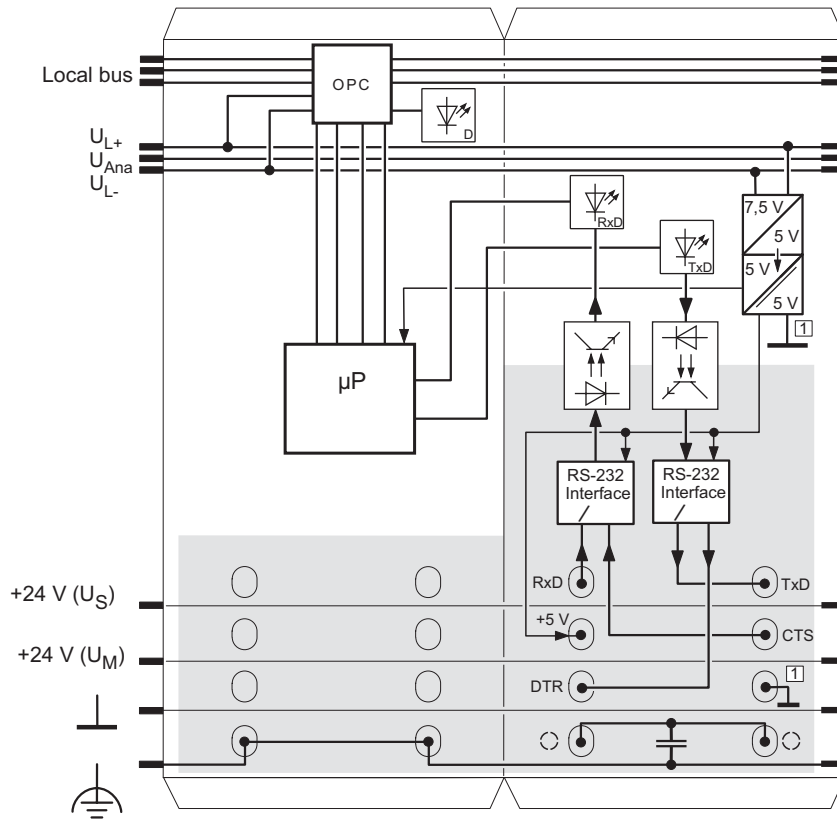
4.3 Terminal point assignment

Connector	Terminal point	Signal	Assignment
1	1.4, 2.4	FE	Functional earth ground
	All other terminal points of this connector are not used.		
2	1.1	RxD	Serial data input
	2.1	TxD	Serial data output
	1.2	+5 V	Control output, internally wired to +5 V DC
	2.2	CTS	Control input for hardware handshake
	1.3	DTR	Control output for hardware handshake
	2.3	GND	GND for serial interface
	1.4, 2.4	Shield	Shield connection



Observe the connection notes on page 9.





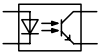




5 Internal circuit diagram



7112A004

Figure 2 Internal wiring of the terminal points

Key:

	Protocol chip (bus logic including voltage conditioning)		Microprocessor
	Diagnostic and status indicators with function information		RS-232 Interface
	Optocoupler		Capacitor
	DC/DC converter with electrical isolation		Ground, electrically isolated from ground of the communications power U_L
	Other symbols used are explained in the IL SYS INST UM E user manual.		

6 V.24 (RS-232) interface



The connector set consists of a shield connector and a standard connector.

The V.24 (RS-232) interface on the terminal represents a piece of DTE (data terminal equipment). This means that connector 2 terminal point 2.1 (TxD) is always used to transmit and connector 2 terminal point 1.1 (RxD) is always used to receive.

The standard requires that DCE (data communication equipment) be connected to the V.24 interface as a peer. However, DTE can also be connected. Please refer to the connection notes under 6.2 and 6.3.

Measuring the voltage between the connection points for the TxD and GND signals in idle state will determine whether the device to be connected to the V.24 (RS-232) interface is a form of DTE or DCE. If the voltage measures approximately **-5 V**, the device is a form of **DTE**. If the voltage is approximately **0 V**, the device is a form of **DCE**.

Example: When using a 25-pos. standard connector (see Figure 3 on page 7) the voltage between **pin 2 (TxD)** and **pin 7 (GND)** must be measured.

6.1 V.24 module handshake signals

Any device with a V.24 (RS-232) interface can be connected to the V.24 (RS-232) interface on the terminal. Both the terminal and the device connected to the V.24 (RS-232) interface can act as a transmitter **and** a receiver for data exchange. As errors can occur during data exchange if both devices transmit or receive simultaneously, the **handshake** is used as a procedure for the mutual signaling of clear to receive and clear to transmit.

The terminal supports DTR and CTS handshake signals. Each uses one wire of the connecting cable.

The connecting signals are described from the point of view of the terminal, i.e., from the point of view of the DTE.

Handshake signals:

Signal	Meaning	Direction
CTS (Clear To Send)	<p>The terminal receives the CTS signal from the connected device via the V.24 (RS-232) interface. If the CTS signal is set to <i>High</i>, the terminal can transmit data.</p> <p> The exception is: 3964R, XON/XOFF Protocol</p> <p> You can disable the CTS signal evaluation (see "Transmission without hardware handshake on page 17").</p>	Input
DTR (Data Terminal Ready)	<p>The DTR signal is transmitted from the IB IL RS 232-PRO terminal, i.e., set to <i>High</i>, once it is ready to receive. The peer connected to the V.24 (RS-232) interface is now able to transmit. After 4,095 characters (4 kbytes), the terminal receive buffer is full and the DTR signal is set to <i>Low</i>. As soon as more characters are read from the bus side, the DTR signal is set to <i>High</i> and the terminal is ready to receive.</p> <p> With the transparent and XON/XOFF protocols, DTR is set to "0" if fewer than 15 characters are free in the receive FIFO.</p>	Output

6.2 V.24 interface wiring with four-wire handshake

The TxD, RxD, DTR, and CTS signals are used for a four-wire handshake connection between the terminal and the device to be connected. Each signal corresponds to one wire in the connecting cable. An Inline male connector is required on the terminal side. A 9-pin or 25-pin socket is required on the opposite side depending on the device to be connected. Both GND pins are also wired.



In Figure 3 and Figure 4 the shield connector is connected on the right-hand side of the terminal. In this case, a capacitor is placed between the shield and FE. If the shield is to be placed directly on FE, the shield connector must be connected on the left-hand side of the terminal. Observe the connection notes on page 9.

In Figure 3 and Figure 4, it is assumed that the signal assignment of the connectors for the device to be connected corresponds to the assignment of a PC connector. In individual cases, however, the signal assignment of the pins might be different, as the DTE-DTE connections as well as the connections between 25-pin and 9-pin connectors and sockets are not standardized.

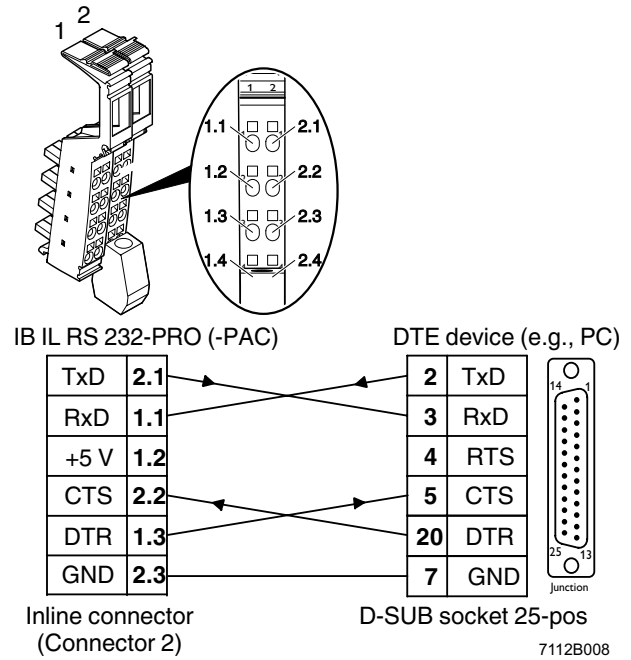


Figure 3 V.24 (RS-232) interface wiring with handshake for DTE (25-pin)

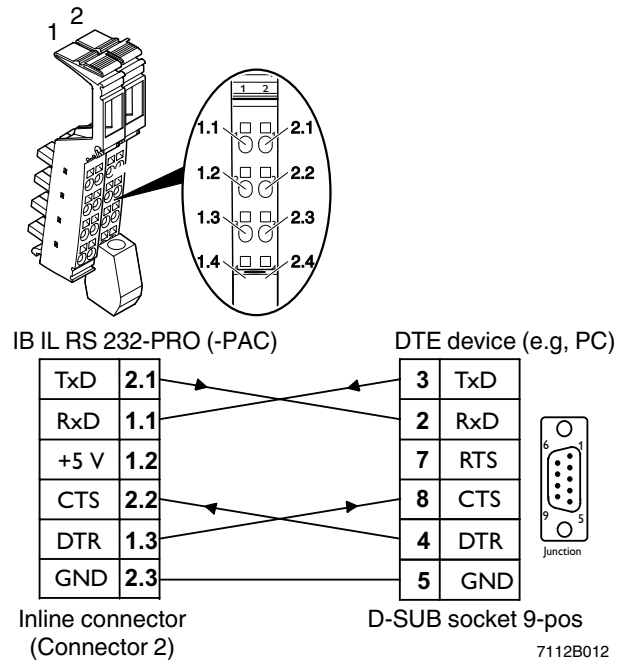


Figure 4 V.24 (RS-232) interface wiring with handshake for DTE (9-pin)

6.3 V.24 (RS-232) interface wiring without handshake

For wiring without handshake, the transmission can only be executed with the help of both TxD and RxD signals. In the same way as the GND contacts. Both wires for the TxD and RxD signals are connected to the IB IL RS 232-PRO terminal male connector and are soldered to the socket on the side of the device to be connected.

In addition, a jumper is connected on the male connector between the terminal points for the +5 V and CTS signals and on the socket between the pins for the RTS and CTS signals.

In this way, permanent readiness to receive of the peer is simulated and the connected device is again able to transmit data via the V.24 interface.



In Figure 5 and Figure 6 the shield connector is connected on the right-hand side of the terminal. In this case, a capacitor is placed between the shield and FE.

If the shield is to be placed directly on FE, the shield connector must be connected on the left-hand side of the terminal. Observe the connection notes on page 9.

In Figure 5 and Figure 6, it is assumed that the signal assignment of the connectors for the device to be connected corresponds to the assignment of a PC connector.

In individual cases, however, the signal assignment of the pins might be different, as the DTE-DTE connections as well as the connections between 25-pin and 9-pin connectors and sockets are not standardized.

The terminal sets the DTR signal to *Low* before the receive FIFO overflows. As the DTR signal is not evaluated for wiring without handshake, some of the data sent to the module via the V.24 interface may be lost until the module is ready to receive again.

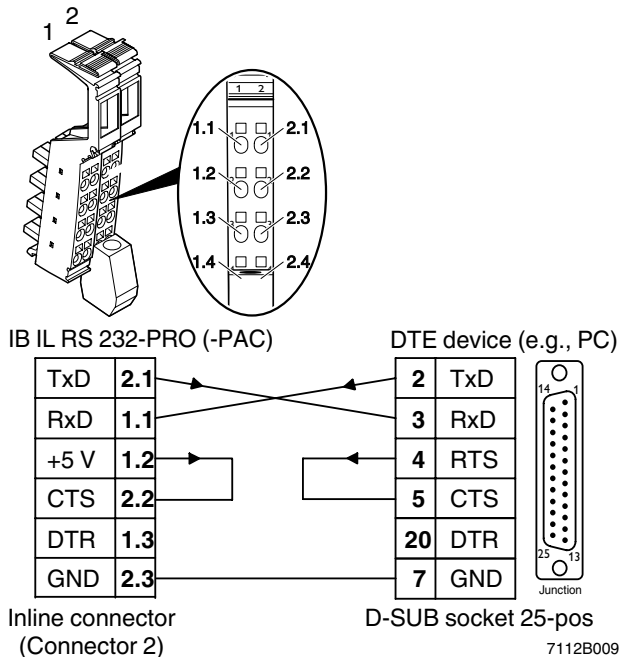


Figure 5 V.24 interface wiring without handshake for DTE (25-pos.)

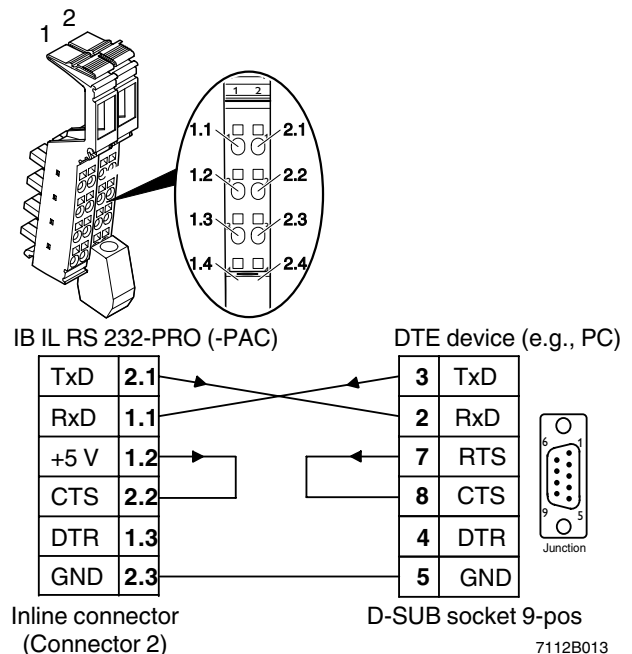


Figure 6 V.24 interface wiring without handshake for DTE (9-pos.)

7 Connection notes



By assigning terminal points 1.4 and 2.4 of both connectors you can connect the cable shield either using a capacitor (connector 2) or directly (connector 1) to functional earth ground (FE).

With the two connection options, you can connect one side of the cable shield directly and one side using a capacitor to FE without any additional effort. In this way, you can prevent ground loops occurring if a shield with two direct connections were placed on FE.

If you connect the shield via connector 1, you must connect the shield connector on the left-hand side of the terminal. All wires must be connected to connector 2.

Ensure that on connector 2, terminal point 1.2 (+5 V) is exclusively used to provide the 5 V signal for the CTS input (terminal point 2.2), in the event of communication without handshake. In this case insert a jumper between the terminal points. Any other use is not permitted.



Use a connector with shield connection when installing the I/O device. Figure 7 and Figure 8 show the connection schematically (without shield connector).

7.1 Shield capacitively connected to FE

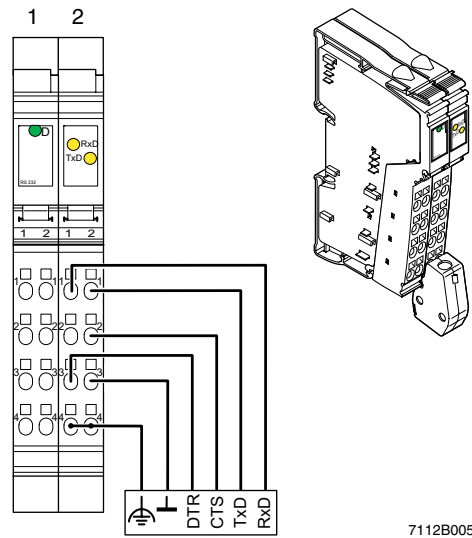


Figure 7 Connection of an I/O device with a serial interface

In this example, the V.24 interface wiring for communication with 4-wire handshake is shown.

7.2 Shield connected directly to FE

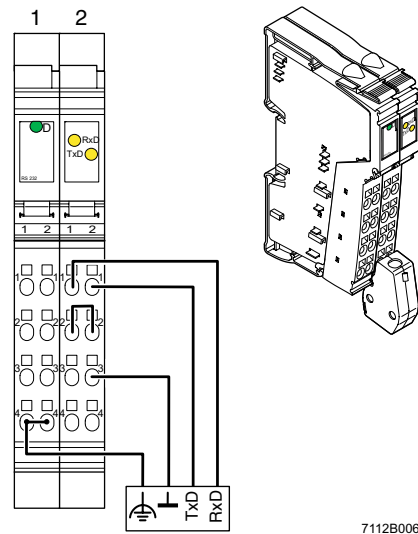


Figure 8 Connection of an I/O device with a serial interface

In this example, the V.24 interface wiring for communication without handshake is shown. You should insert a jumper between connection points 1.2 (+5 V) and 2.2 (CTS).

8 Data storage and transmission

The terminal stores the received serial data in an intermediate buffer until it is requested from the serial interface by the bus controller board or the device. Serial data traffic can be managed using various protocols. The protocol used depends on the type of protocol supported by the peers.

8.1 Overview of supported protocols

Protocol	Receive buffer	Transmit memory	Special features when receiving
Transparent	4096 bytes	1023 bytes	
End-to-end	3 buffers with 330 bytes each	1023 bytes (including end characters)	Two end characters are filtered out
Dual buffer	2 buffers with 330 bytes each	1023 bytes (including end characters)	Only stores the most recently received data, end characters are filtered out
3964R	3 buffers with 330 bytes each	3 buffers with 330 bytes each	Data exchange with software handshake, time monitoring, and checksum
XON/XOFF	4096 bytes	1023 bytes	Software handshake

8.2 Transparent protocol

If the transparent protocol is used, serial data is transmitted through the terminal in the same format it was received from the serial interface or the bus side.

The transmit FIFO (First-In-First-Out memory) can store 1023 bytes (1 kbyte) and the receive FIFO can store 4096 bytes (4 kbytes). If the terminal receives another character after the 4095th character, the error pattern is stored in the receive FIFO. All further characters are ignored.

8.3 End-to-end protocol

The serial data is conditioned for the end-to-end protocol.

If serial data is sent from the bus side, two additional characters, the first and second delimiters, are attached for transmission to the serial interface. The first and second delimiters are defined upon terminal configuration.

Serial data sent from the serial interface can only be read by the user if the terminal has received the first and second delimiters. The two delimiters confirm that the serial data has been received without error and the maximum data length of 330 bytes has been observed. The delimiters are filtered out when the data is read by the bus side.

Unlike in the transparent protocol, the receive memory is not organized as a FIFO but as a buffer. 3 buffers with 330 bytes each are available. If the buffer size of 330 bytes is exceeded, without the two delimiters being detected, the buffer is overwritten again.

The transmit FIFO consists of 1023 bytes. The delimiters are attached to, and stored with, the data to be sent.

8.4 Dual buffer protocol

With this protocol, the **last** received data block is stored. A data block is defined as a sequence of characters with the first and second delimiter, as in the end-to-end protocol.

As soon as a new data block is received, the previous one is overwritten. This is achieved by means of two buffers, which are written alternately. In this way, there is always one buffer ready to receive serial data while the second buffer stores the most recently received data block. A data block is only regarded as complete once both delimiters have been detected, one after the other. It can then be read from the bus side.

If the buffer size of 330 bytes is exceeded, without the two delimiters being detected, the buffer is overwritten again.

When transmitting serial data, the same is valid as for the end-to-end protocol: If serial data is sent from the bus side, two additional characters, the first and second delimiters, are attached for transmission to the serial interface.

8.5 3964R protocol

This protocol, developed by Siemens, is the most complex. It uses beginning and end identifiers, checksum and a time monitoring function.

3 buffers are available for transmitting, 3 buffers are available for receiving.

Character delay time: 220 ms
 Acknowledgment delay time: 2 s
 Block waiting time: 10 s
 Number of attempts to establish a connection: 6

The optional 3964 priority defines which device may send first (high priority) if there is an initialization conflict (several devices attempting to send data simultaneously).

8.6 XON/XOFF protocol

This protocol operates like the transparent protocol, however, using a software handshake.

Data transmission with this protocol is controlled by the XON and XOFF characters. XON is set to 11_{hex} and XOFF to 13_{hex}.

If the terminal receives an XOFF, no more serial data will be sent until an XON is received.

The terminal itself will transmit an XOFF if the available space in the receive memory is less than 15 bytes. As soon as more memory becomes available again, the module will transmit a single XON.

Serial data is not filtered when it is transmitted. So any characters occurring with the code defined for XON and XOFF are transmitted and may trigger undesirable events at the receiver. When serial data is received, the XON and XOFF characters are filtered and are not available as data. Any characters with the XON or XOFF code are lost. Ensure that characters with these codes do not appear in the data stream.

9 Programming data/ configuration data

9.1 Local bus (INTERBUS)

ID code	BF _{hex} (191 _{dec})
Length code	06 _{hex}
Process data channel	96 bits
Input address area	12 bytes
Output address area	12 bytes
Parameter channel (PCP)	0 bytes
Register length (bus)	12 bytes

9.2 Other bus systems



For the programming/configuration data of other bus systems, please refer to the corresponding electronic device data sheet (e.g., GSD, EDS).

10 Process data

The terminal process image comprises six data words each in input and output direction.



For the assignment of the illustrated (word.bit) or (byte.bit) view to your **INTERBUS** control or computer system, please refer to the DB GB IBS SYS ADDRESS data sheet.

The terminal has six process data words.

Word	0		1		2		3		4		5	
Byte in the Motorola format	0	1	2	3	4	5	6	7	8	9	10	11
Byte in the Intel format	1	0	3	2	5	4	7	6	9	8	11	10
OUT	Command/parameter	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data
IN	Status parameter	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data



The byte representation in the Motorola format, also called Big Endian (high order byte at starting address) corresponds to the INTERBUS standard representation. All byte representations in the data sheet have this format.

The byte representation in the Intel format is also called Little Endian (low order byte at starting address).

The command is used to determine the function. The actually transmitted data depends on the command.

10.1 Word 0 general

Control word

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Command	OUT parameter						x	x	x	x	x	x	x	x

Status word

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Err	Command	IN parameter						x	x	x	x	x	x	x	x

Err: error

x = 0 or 1; the assignment depends on the command.



Bit 15 = error: Is not valid for the Read Characters command. For the meaning of bit 15 for this command, please refer to Section "Read Characters command" on page 16.

Code (Bin)	Code (hex) (With bit 15 = 0)	Command
000	0	Read status bits. Input word 1 contains the number of characters received.
001	1	Transmit characters
010	2	Store characters temporarily
011	3	Read characters OUT parameter = C _{hex} : Read FW version, OUT parameter = D _{hex} : Read configuration
100	4	Write configuration
101	5	Toggle command 1: Transmit characters
110	6	Toggle command 2: Store characters temporarily
111	7	Toggle command 3: Read characters

Command toggling

Command toggling is used to execute a command on a terminal again. In this way a second command code is available for the same function. This applies for the following commands:

- Transmit characters
- Store characters temporarily
- Read characters

Here, bit 14 is used for toggling. If, for example, you wish to transmit character strings in sequence, use the command code 001_{hex} for the first transmission and then use 101_{hex} and 001_{hex} alternately.

10.2 Read Status Bits command

Format of the process data word 0

Control word

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0					DTR					Transmit error	Receive error	

	DTR (if DTR control enabled)
Code	Meaning
0 _{hex}	DTR = Logic 0
1 _{hex}	DTR = Logic 1

	Receive error
Code	Meaning
0 _{hex}	No action
1 _{hex}	Reset receive error



The DTR signal can only be controlled when DTR control is enabled via process data, that is when the DTR control bit has been enabled in the configuration (see "Write Configuration command" on page 17).

	Transmit error
Code	Meaning
0 _{hex}	No action
1 _{hex}	Reset transmit error

Status word

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	Amount of received data			CTS	Receive buffer not empty	Transmit buffer full	Receive buffer full	Res.	Transmit error	Receive error	Receive buffer not empty	

	CTS (if configured)
Code	Meaning
0 _{hex}	CTS = Logic 0
1 _{hex}	CTS = Logic 1

	Receive buffer full
Code	Meaning
0 _{hex}	Not full
1 _{hex}	Full



The CTS bit is only updated when the "Mapping the CTS signal" bit was set in the configuration (see "Write Configuration command" on page 17).

	Transmit error
Code	Meaning
0 _{hex}	No error
1 _{hex}	Error

	Send buffer not empty
Code	Meaning
0 _{hex}	Empty
1 _{hex}	Not empty

	Receive error
Code	Meaning
0 _{hex}	No error
1 _{hex}	Error

	Send buffer full
Code	Meaning
0 _{hex}	Not full
1 _{hex}	Full

	Receive buffer not empty
Code	Meaning
0 _{hex}	Empty
1 _{hex}	Not empty

Bit/status	Effect	Protocol
Bit 0 = '1'	The receive buffer is not empty, there are characters to be read.	All
Bit 1 = '1'	The receive error indicates that a 3964R telegram could not be received without error after six transmit attempts by the serial peer or after the block waiting time had elapsed.	3964R
Bit 2 = '1'	The transmit error indicates that a 3964R telegram could not be transmitted from the module to the serial peer without error after six transmit attempts. The telegram was rejected.	3964R
Bit 3 = '1'	Reserved	
Bit 4 = '1'	The receive buffer is full: Transparent and XON/XOFF protocol: Residual capacity: < 15 characters 3964R and end-to-end protocol: Residual capacity: none	Transparent, end-to-end, 3964R, XON/XOFF
Bit 5 = '1'	The transmit buffer is full: 3964R protocol: Residual capacity: none Dual buffer, transparent, end-to-end, and XON/XOFF protocol: Residual capacity: ≤ 30 characters	All
Bit 6 = '1'	The transmit buffer is not empty, characters to be transmitted are available.	All
Bit 7 = '1'	The CTS status is mapped here when the "Mapping the CTS signal" bit was set in the configuration.	All
Bits 8 to 11	Number of characters received. If the code = F_{hex} , more than 14 characters have been received.	



Both error bits (bits 1 and 2) are not automatically reset. They can only be reset by the process data output word.



In the transparent and XON/XOFF protocols, the input word 1 contains the total number of characters received.

With the Read Status Bits command the content of the input data is continuously updated. Unlike with other commands toggling is not required.

10.3 Transmit Characters command

Process data is stored in the transmit memory and then directly transmitted via V.24 (RS-232). A maximum of eleven characters can be transmitted. The OUT parameter determines the number of characters to be transmitted.

Characters stored in the intermediate buffer are transmitted first. After the command has been executed successfully the intermediate buffer is cleared.

Process data assignment for the Transmit Characters command with 17 characters (C1 - C17)

Word	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	1B _{hex}	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
IN	1B _{hex}	–	–	–	–	–	–	–	–	–	–	–
OUT	56 _{hex}	C12	C13	C14	C15	C16	C17	–	–	–	–	–
IN	56 _{hex}	–	–	–	–	–	–	–	–	–	–	–

For protocols with end identifiers the end identifier is added after every transmitted block (11 characters, maximum)

Reasons for an error bit set:

- OUT parameter = 0 **and** intermediate buffer empty
- OUT parameter >11
- Not enough space in the transmit memory
- Not enough space in the intermediate buffer

10.4 Store Characters Temporarily command

The transmit data is stored in an intermediate buffer, which can store 330 characters. No characters are transmitted. The OUT parameter determines the number of characters. The Transmit Characters command is used for transmitting

the data stored temporarily. In this way character blocks of up to 330 characters can be transmitted. They are divided over 20 telegrams with 11 characters each.

Process data assignment for the Save Characters command and subsequent Transmit Characters command with 41 characters (C1 - C41).

Word	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	2B _{hex}	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
IN	2B _{hex}	–	–	–	–	–	–	–	–	–	–	–
OUT	6B _{hex}	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22
IN	6B _{hex}	–	–	–	–	–	–	–	–	–	–	–
OUT	2B _{hex}	C23	C24	C25	C26	C27	C28	C29	C30	C31	C32	C33
IN	2B _{hex}	–	–	–	–	–	–	–	–	–	–	–
OUT	18 _{hex}	C34	C35	C36	C37	C38	C39	C40	C41	–	–	–
IN	18 _{hex}	–	–	–	–	–	–	–	–	–	–	–

For protocols with end identifiers the end identifier is added after every transmitted block (41 characters)

Reasons for an error bit set:

- OUT parameter = 0 **or** >11
- Not enough space in the intermediate buffer

10.5 Read Characters command

This command is used to read a maximum of eleven characters. The IN parameter contains the number of valid characters available in the input data.

Process data assignment for the Read Characters command with eleven characters (C1 - C11)

Word	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	30 _{hex}	–	–	–	–	–	–	–	–	–	–	–
IN	3B _{hex}	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11

For the end-to-end, dual buffers and 3964R protocols, bit 15 in the status word is not used as error bit but indicates whether there are still other characters from the received block that remain to be read.

For the indicated protocols, a data transfer is handled as block that is identified by a special end identification. When reading the characters received, it is important to know whether the characters provided with this command all belong to one single block or whether there are still characters that remain to be read. The status bit "Receive buffer is not empty" does not provide this information as it also indicates whether additional blocks have been received. For this reason, bit 15 is used in this case. A maximum block length of 330 characters is supported.

Bit 15 in the status word

Bit/ status	Effect	Protocol
Bit 15 = '0'	The characters read are the last ones in the received block.	End-to-end, dual buffer, 3964R
Bit 15 = '1'	There are still some more characters to be read from the received block.	



Together with the toggle bit, the upper nibble of the status word might have the values B_{hex} (1011_{bin}) and F_{hex} (1111_{bin}).

Example: In the 3964R protocol a block with 20 characters was received. For reading, the Read Characters command must be given twice. The second time the command is given, the toggle bit is set.

Process data assignment for the Transmit Characters command with 20 characters (C1 - C20)

Word	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	30 _{hex}	–	–	–	–	–	–	–	–	–	–	–
IN	BB _{hex}	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
OUT	70 _{hex}	–	–	–	–	–	–	–	–	–	–	–
IN	79 _{hex}	C12	C13	C14	C15	C16	C17	C18	C19	C20	–	–

10.6 Write Configuration command

Process data assignment for the Write Configuration command

Output words 0 to 5

Word	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	40 _{hex}	Error pattern	Protocol	Baud rate data width	1st delimiter	2nd delimiter	Direct baud rate	Reserved	Reserved		Reserved	
IN	40 _{hex}	-	-	-	-	-	-	-	-	-	-	-

Output word 1 for the Write Configuration command

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Transmission w/o HW handshake		DTR control	Mapping the CTS signal	Protocol				Baud rate				Data width			

Element value range



The options in **bold** are default settings.

Transmission without hardware handshake		
Code	Meaning	Representation in CMD/PC WORX
0 _{hex}	Transmission dependent on CTS signal Exception: 3964R, XON/XOFF Protocol CTS = 0: Transmission disabled, CTS = 1: Transmission enabled	<i>Transmission dependent on CTS signal</i>
1 _{hex}	Transmission independent of CTS signal	<i>Transmission independent of CTS signal</i>

DTR control		
Code	Meaning	Representation in CMD/PC WORX
0 _{hex}	Automatic	<i>Automatic</i>
1 _{hex}	Via process data	<i>Via process data</i>

CTS signal mapping		
Code	Meaning	Representation in CMD/PC WORX
0 _{hex}	CTS signal is not mapped to input process data	<i>CTS signal is not mapped to input process data</i>
1 _{hex}	CTS signal is mapped to input process data	<i>CTS signal is mapped to input process data</i>

	Protocol
Code	Meaning
00_{hex}	Transparent
01 _{hex}	End-to-end
02 _{hex}	Dual buffer
03 _{hex}	3964R with low priority
04 _{hex}	3964R with high priority
05 _{hex}	XON/XOFF

	Baud rate
Code	Value
00 _{hex}	110 baud
01 _{hex}	300 baud
02 _{hex}	600 baud
03 _{hex}	1200 baud
04 _{hex}	1800 baud
05 _{hex}	2400 baud
06 _{hex}	4800 baud
07_{hex}	9600 baud
08 _{hex}	19200 baud
09 _{hex}	38400 baud
0D _{hex}	Directly, basis 500 kbaud
0E _{hex}	Directly, basis 62.5 kbaud
0F _{hex}	Directly, basis 15625 baud



The specified baud rates of 110 baud to 38400 baud are adequate for most applications. However, you can freely choose the baud rate by direct programming. For this, use the 0D_{hex}, 0E_{hex}, and 0F_{hex} baud rate codes in output word 1. See "Direct Baud Rate (DBC)" on page 20.

	Data width			
Code	Meaning			Representation in CMD/PC WORX
	Data bits	Parity	Stop bits	
00 _{hex}	7	Even	1	7 data bits, even parity, 1 stop bit
01 _{hex}	7	Odd	1	7 data bits, odd parity, 1 stop bit
02_{hex}	8	Even	1	8 data bits, even parity, 1 stop bit
03 _{hex}	8	Odd	1	8 data bits, odd parity, 1 stop bit
04 _{hex}	8	None	1	8 data bits, without parity, 1 stop bit
05 _{hex}	7	None	1	7 data bits, without parity, 1 stop bit
06 _{hex}	7	Even	2	7 data bits, even parity, 2 stop bits
07 _{hex}	7	Odd	2	7 data bits, odd parity, 2 stop bits
08 _{hex}	8	Even	2	8 data bits, even parity, 2 stop bits
09 _{hex}	8	Odd	2	8 data bits, odd parity, 2 stop bits
0A _{hex}	8	None	2	8 data bits, without parity, 2 stop bits
0B _{hex}	7	None	2	7 data bits, without parity, 2 stop bits

Error pattern	
Code	Meaning
24_{hex}	\$
XX _{hex}	Any character

First delimiter	
Code	Meaning
0D_{hex}	Carriage Return (CR)
XX _{hex}	Any character

Second delimiter	
Code	Meaning
0A_{hex}	Line Feed (LF)
XX _{hex}	Any character

The **error pattern** contains the character that is written into the FIFO, if a character was received with an error (not valid for the 3964R protocol). Reasons are e.g. parity errors, exceeded value range, noise interference. In the transparent and XON/XOFF protocols, the pattern is also used if the receive FIFO is full and further characters are received.

The **first delimiter** and the **second delimiter** contain the end characters for the dual buffer and the end-to-end protocols.



After successful configuration the characters for the receive and transmit FIFO are reset. In this way, all transmit and receive data that have not yet been processed will be deleted.

Reasons for an error bit set:

- Using a reserved code
- Setting a reserved bit

Example

- Default: Transparent protocol
- Baud rate: 19200 baud
- Data width: 8 data bits with odd parity and one stop bit
- Configuration (in hex): 4000 0083 0000 0000

Word	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
Meaning	40 _{hex}	Error pattern	Protocol	Baud rate data width	1st delimiter	2nd delimiter	Direct baud rate	Reserved	Reserved		Reserved	
OUT	40	00	00	83	00	00	00	-	-	-	-	-
IN	40	-	-	-	-	-	-	-	-	-	-	-

Direct Baud Rate (DBC)

Direct programming of the baud rate is selected in the output word 1 using the 0D_{hex}, 0E_{hex} and F_{hex} baud rate codes. You can select a basic clock for the baud rate. The actual baud rate is calculated according to the following formula:

$$\text{Baud rate} = \text{basic clock} / (\text{DBC} + 1)$$

Specify DBC in the output byte 6. To determine DBC change the equation to read:

$$\text{DBC} = \text{basic clock} / \text{baud rate} - 1$$

Example:

The baud rate is 15625 baud. A basic baud rate of 500 kbaud (code 0D_{hex}) is chosen. Determine the direct baud rate:

$$\begin{aligned} \text{DBC} &= (500000 \text{ baud} / 15625 \text{ baud}) - 1 \\ &= 31_{\text{dec}} = 1F_{\text{hex}} \end{aligned}$$

Here is an example for the first 4 output words: 40 00 00 D2 00 00 1F_{hex}.



Programming of the direct baud rate theoretically enables a maximum value of 500 kbaud. Proper operation of the terminal is tested and guaranteed for up to 38400 baud. Operation with higher baud rates depends on the application.

10.7 Read Configuration command

Process data assignment for the Read Configuration command

Word	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	3D _{hex}	x	x	x	x	x	x	x	x	x	x	x
IN	3D _{hex}	Error pattern	Protocol	Baud rate data width	1st delimiter	2nd delimiter	Direct baud rate	00	00	00	00	00

10.8 Read Firmware Version command

With a control word 3C00_{hex} the second input word supplies the firmware version and the type code.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Firmware version, e.g., 100 _{hex}												Type code: 8 _{hex}			

Type code 8_{hex} is identical to the type code of the IB IL RS 232 (-PAC).